JavaScript

Module 7

Ryan Collins

WHAT IS JAVASCRIPT?

A HIGH LEVEL DEFINITION

JavaScript is a scripting programming language that allows you to enable complex features on web pages. A site that has timely content updates, interactive maps or animated 2D/3D graphics, etc. These functionalities are made possible by JavaScript.

JavaScript is a lightweight interpreted programming language. The web browser receives the JavaScript code in its original text form and runs the script from that.

WHAT ARE WE GOING TO MAKE?

FIRST WE HAVE TO LEARN A FEW THINGS

NAMING CONVENTIONS

- Variable and function names use camelCase
- Global variables are written in UPPERCASE
- Constants (const PI = 3.14) in UPPERCASE
- HTML5 attributes start with data (data-quantity)
- CSS properties use hyphens (font-size)
- Underscores (date_of_birth)

Do not start names with a $ sign, as it will conflict with the JavaScript library names

CODE FORMAT

Basics

JavaScript borrows most of its syntax from Java, C, and C++, but it has also been influenced by Awk, Perl, and Python.

JavaScript is **case-sensitive** and uses the **Unicode** character set. For example, the word Früh (which means "early" in German) could be used as a variable name.

```
const Früh = "foobar";
```

But, the variable früh is not the same as Früh because JavaScript is case sensitive.

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Grammar_and_types#basics

COMMENTS

Just like in our HTML and CSS, JavaScript has comments, the syntax of comments is the same as in C++ and many other languages

```
// a one line comment


/* this is a longer,
 * multi-line comment
 */
```

HOW DO WE WRITE JAVASCRIPT?

ADD INLINE JAVASCRIPT TO A WEBPAGE

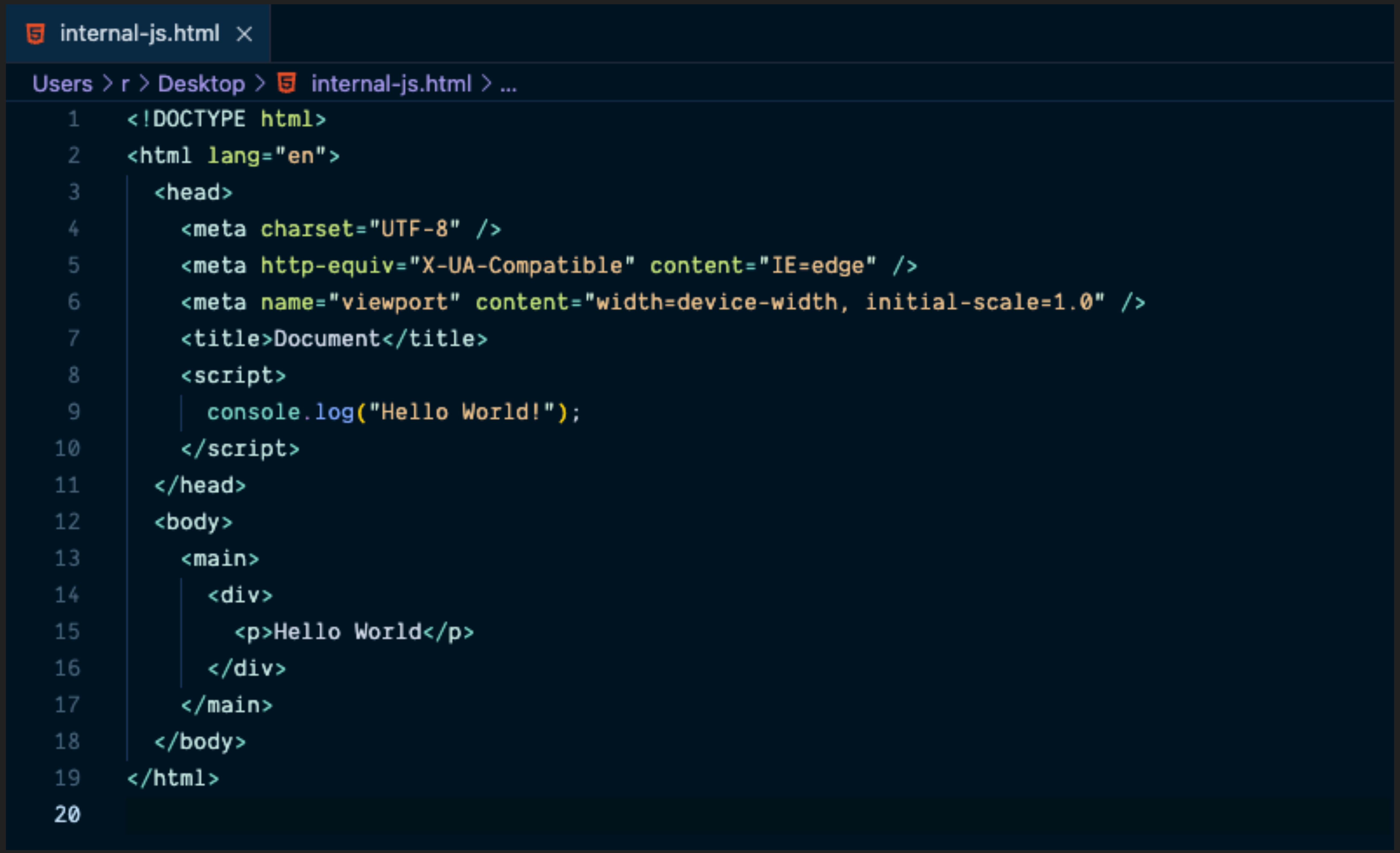
In-line JavaScript places the code directly in the HTML document between two <script> tags. The first <script> tag denotes where the script will be executed.

```
<script type=“text/javascript”>

alert(“Hello world!”)

</script>
```

Copy the code into a HTML file and open it in the browser

INTERNAL JAVASCRIPT

Internal JavaScript add additional elements inside the <html> - <head> and <body> tags.



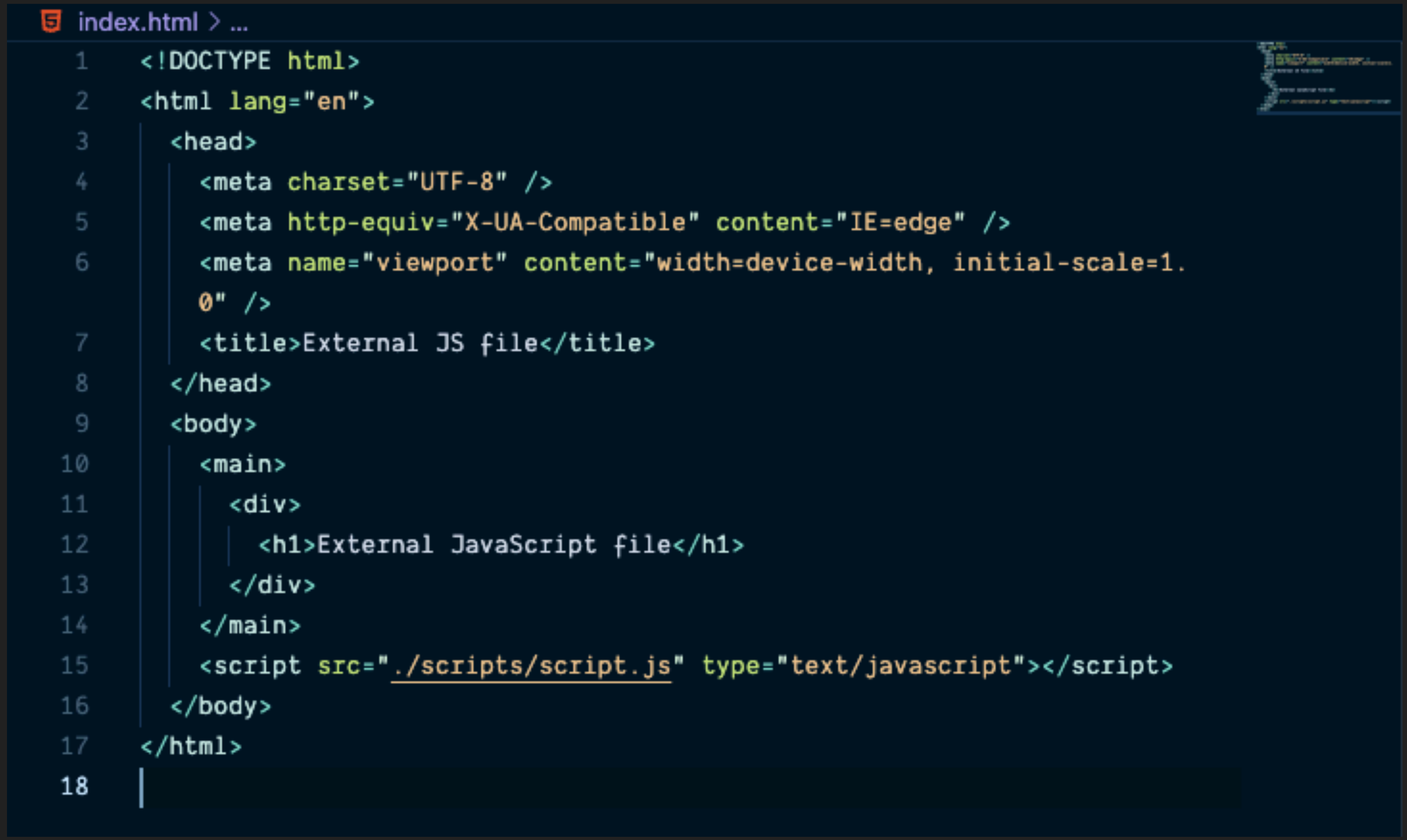
```html
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Document</title>
    <script>
      console.log("Hello World!");
    </script>
  </head>
  <body>
    <main>
      <div>
        <p>Hello World</p>
      </div>
    </main>
  </body>
</html>

```

Check examples folder

EXTERNAL JAVASCRIPT

Linking to an external JavaScript file, just like we do with the CSS stylesheets is considered the best practice for adding JavaScript functionality to your webpage.

Create a file with the postfix .js and add it as a link to your HTML at the end, just before the last closing <body> tag.

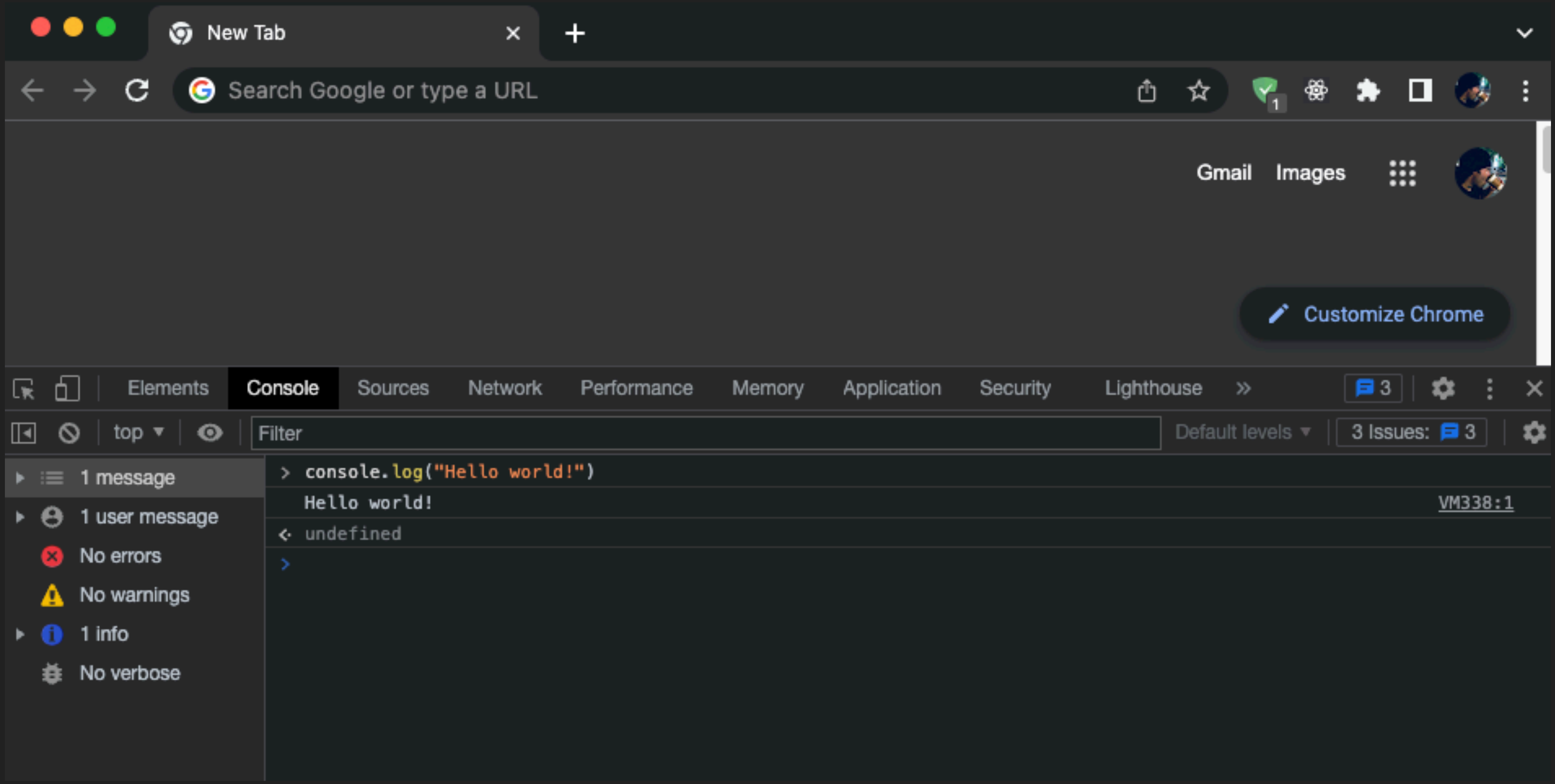


```html
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>External JS file</title>
  </head>
  <body>
    <main>
      <div>
        <h1>External JavaScript file</h1>
      </div>
    </main>
    <script src="./scripts/script.js" type="text/javascript"></script>
  </body>
</html>
```

Check examples folder

JAVASCRIPT IN THE BROWSER

Another way we can write or test JavaScript, is right in the browser developer tools. Open the developer tools and select the Console tab, type in som JavaScript

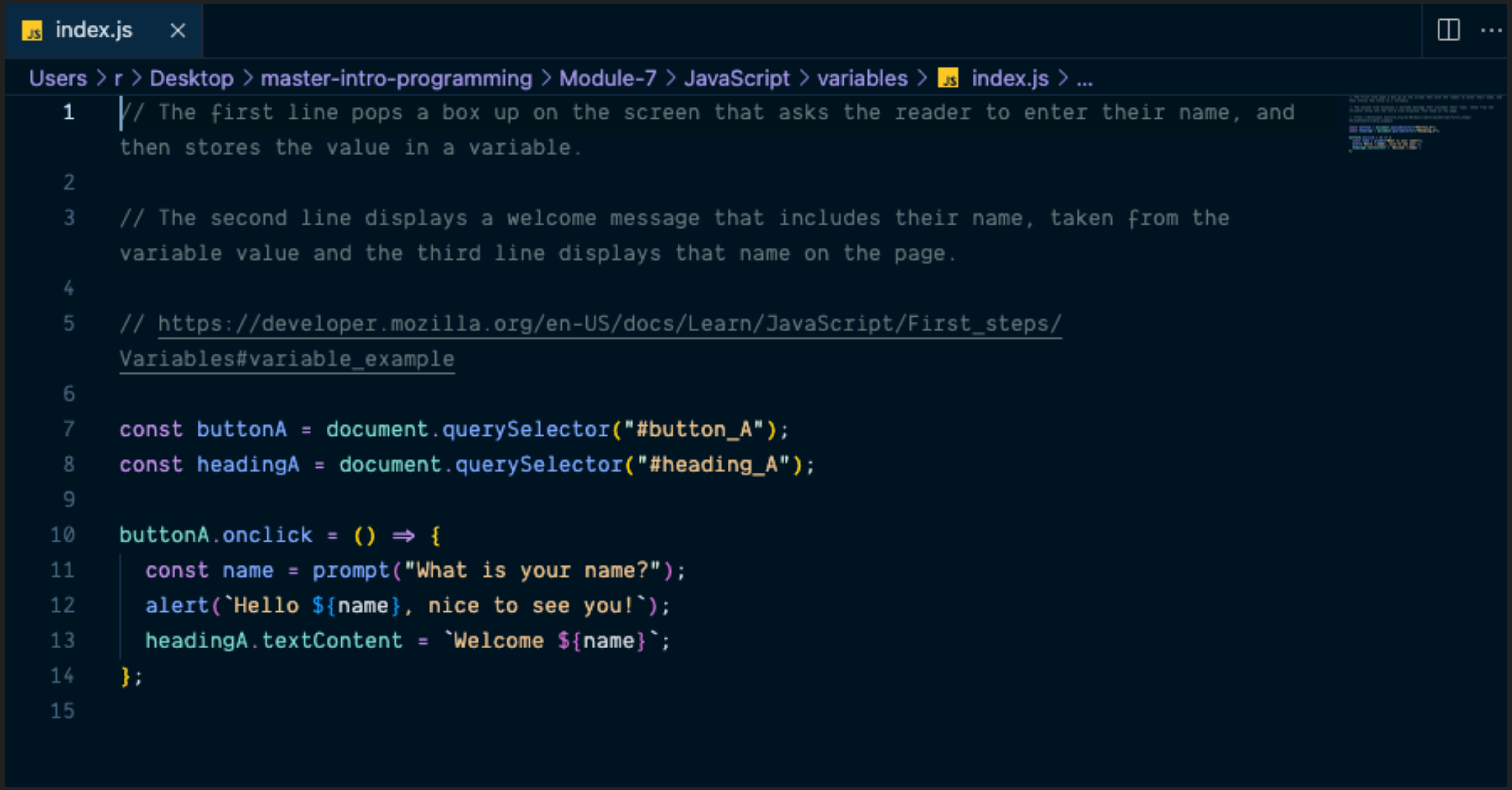


```
> console.log("Hello world!")
  Hello world!                                   VM338:1
<· undefined
>
```

JAVASCRIPT BASICS

VARIABLES

From this example we take the users name that they have entered at the prompt, stored that name in a variable, then later display the stored name back to the user. This is the basic function of a variable. Give it a try in the source code

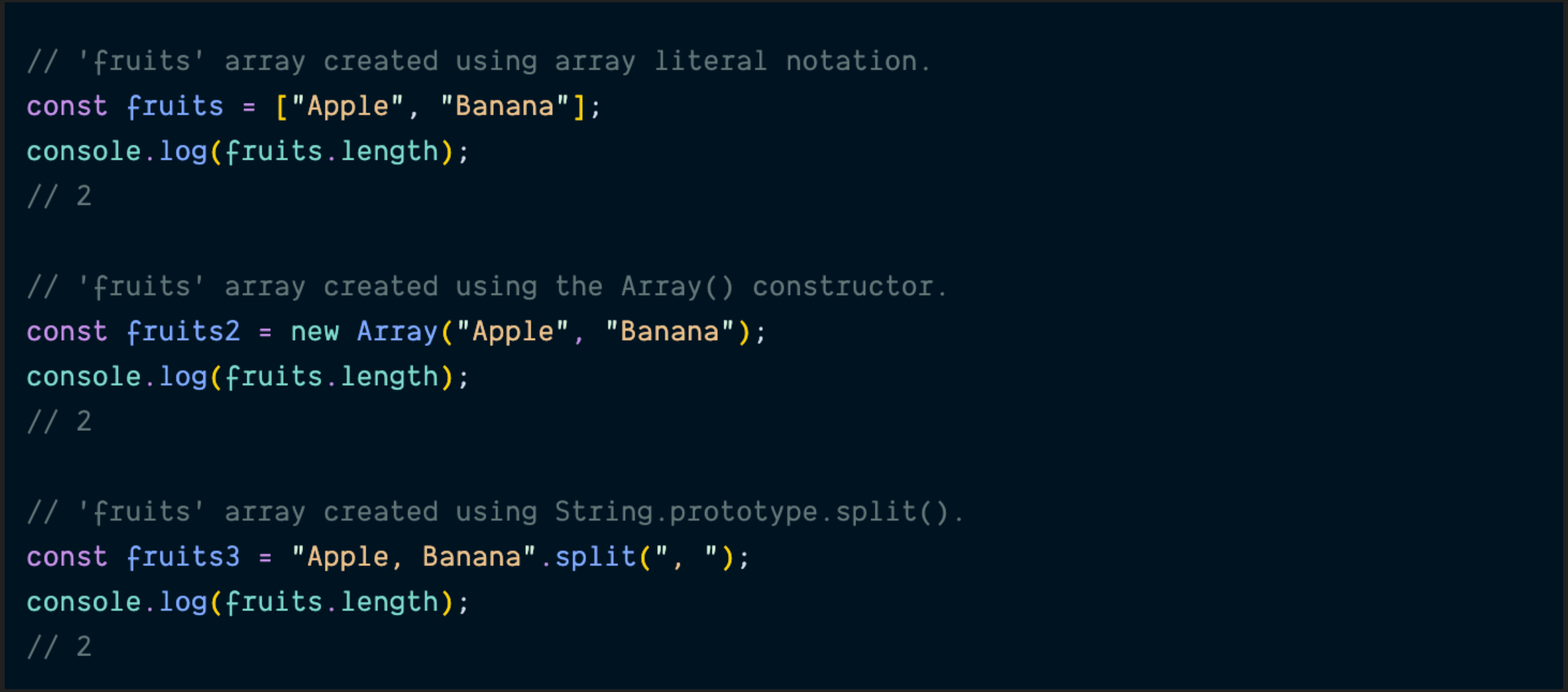

index.js
Users > r > Desktop > master-intro-programming > Module-7 > JavaScript > variables > index.js > ...

```javascript
// The first line pops a box up on the screen that asks the reader to enter their name, and then stores the value in a variable.

// The second line displays a welcome message that includes their name, taken from the variable value and the third line displays that name on the page.

// https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/Variables#variable_example

const buttonA = document.querySelector("#button_A");
const headingA = document.querySelector("#heading_A");

buttonA.onclick = () => {
  const name = prompt("What is your name?");
  alert(`Hello ${name}, nice to see you!`);
  headingA.textContent = `Welcome ${name}`;
};

```

Open index.html from folder named variables

ARRAYS

Arrays basically store a collection of multiple items under a single variable name, like a package of Skittles ;-)
Arrays can also contain a mix of different datatypes.



```
// 'fruits' array created using array literal notation.
const fruits = ["Apple", "Banana"];
console.log(fruits.length);
// 2

// 'fruits' array created using the Array() constructor.
const fruits2 = new Array("Apple", "Banana");
console.log(fruits.length);
// 2

// 'fruits' array created using String.prototype.split().
const fruits3 = "Apple, Banana".split(", ");
console.log(fruits.length);
// 2
```

Run the example index.js with node in VSCode

Node.js® is a JavaScript runtime built on Chrome's V8 JavaScript engine.

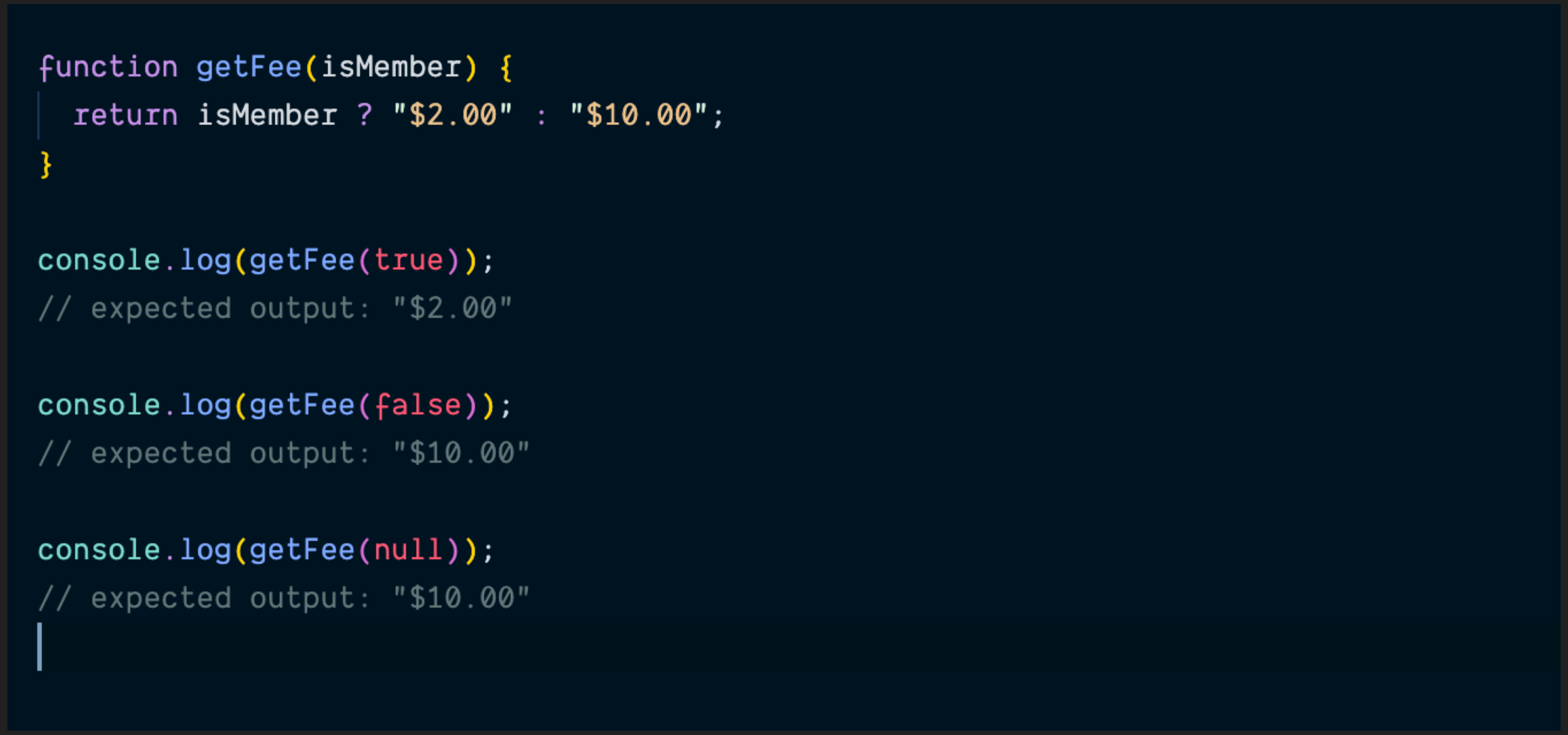
https://nodejs.org/en/

CONDITIONALS

The conditional (ternary) operator is the only JavaScript operator that takes three operands:

a condition followed by a question mark (?), then an expression to execute if the condition is true followed by a colon (:), and finally the expression to execute if the condition is false.

condition ? exprIfTrue : exprIfFalse



```
function getFee(isMember) {
  return isMember ? "$2.00" : "$10.00";
}

console.log(getFee(true));
// expected output: "$2.00"

console.log(getFee(false));
// expected output: "$10.00"

console.log(getFee(null));
// expected output: "$10.00"
```

Run the example index.js with node in VSCode

OBJECTS

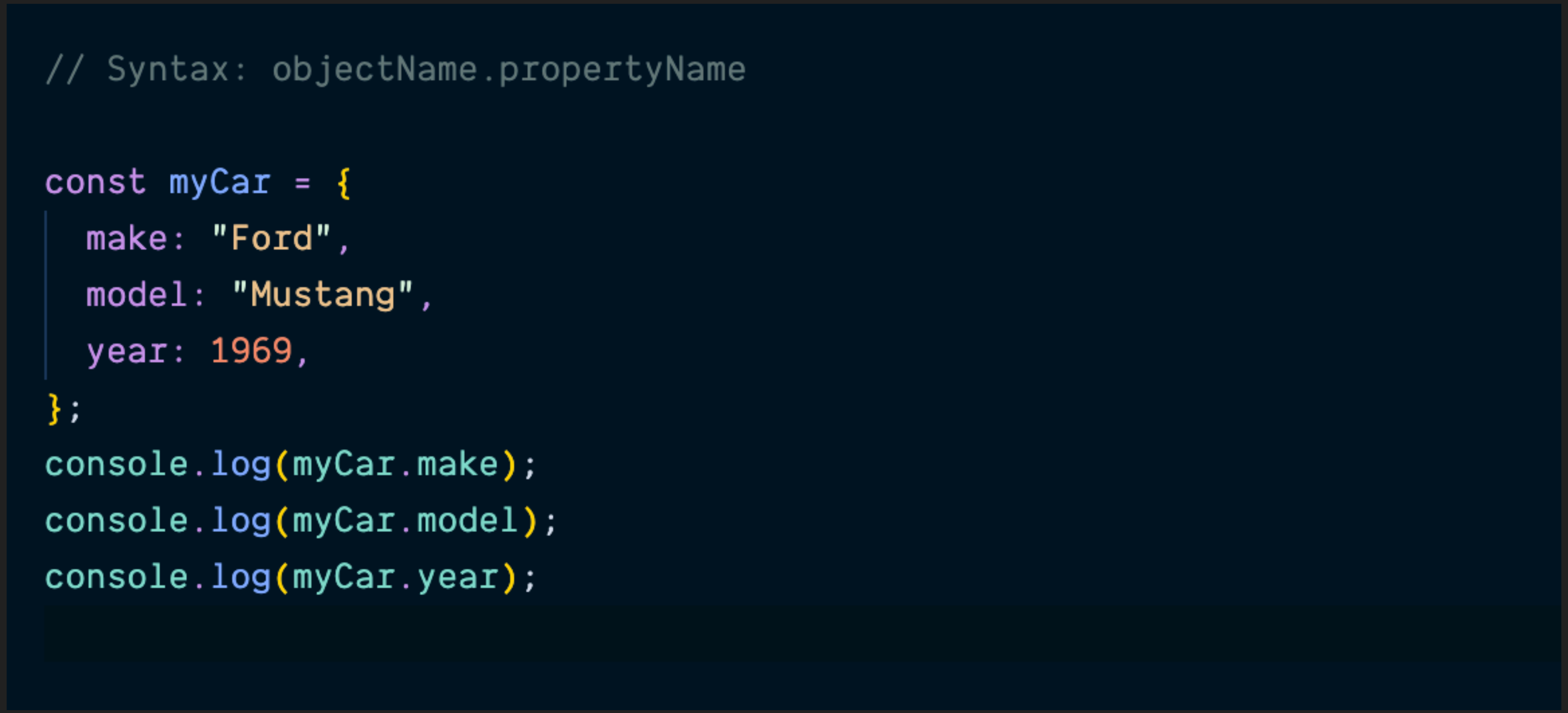
A JavaScript object has properties associated with it. A property of an object can be explained as a variable that is attached to the object.

Object properties are basically the same as ordinary JavaScript variables, except for the attachment to objects.

The properties of an object define the characteristics of the object.

You access the properties of an object with a simple dot-notation:

Syntax: objectName.propertyName



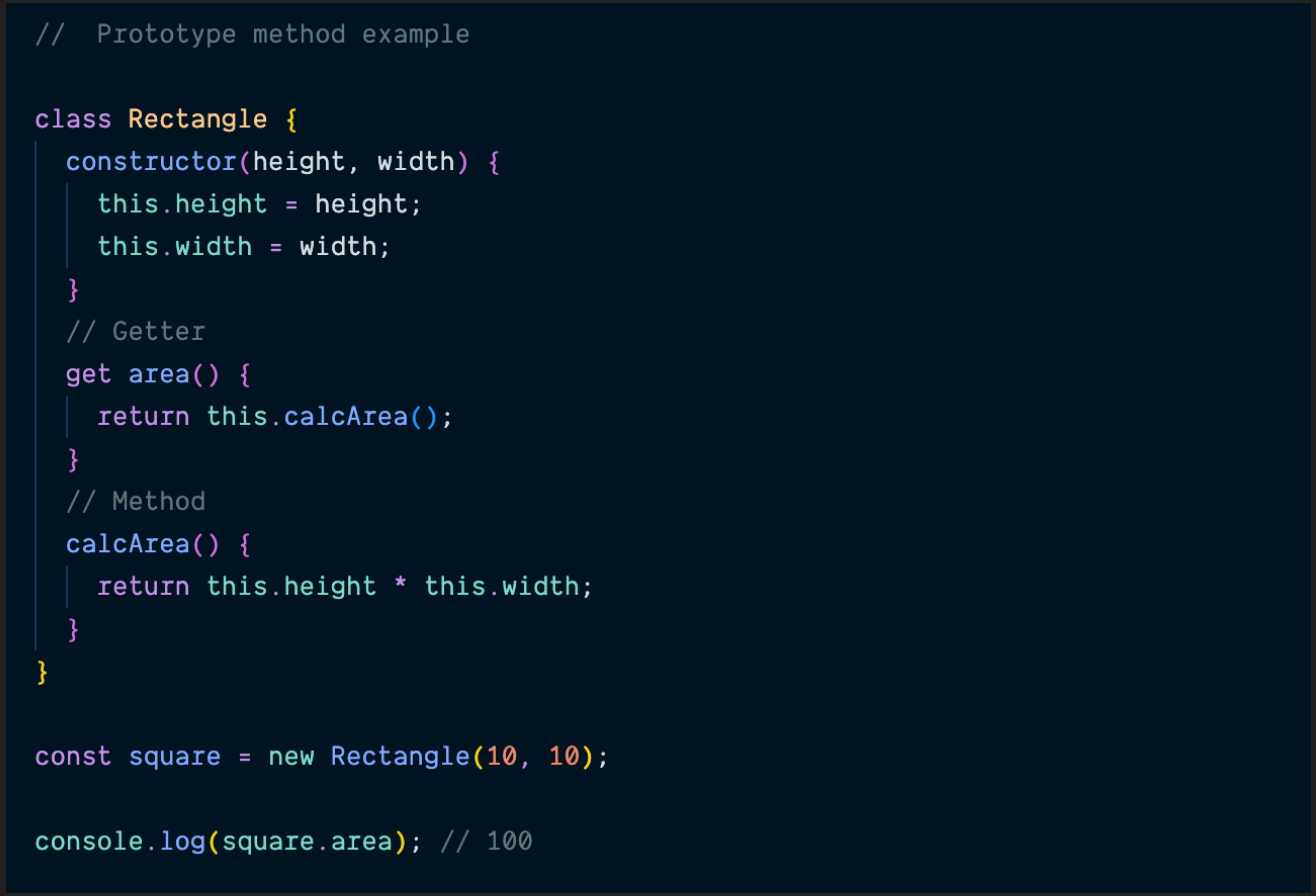
```
// Syntax: objectName.propertyName

const myCar = {
  make: "Ford",
  model: "Mustang",
  year: 1969,
};
console.log(myCar.make);
console.log(myCar.model);
console.log(myCar.year);
```

Run the example index.js with node in VSCode

BASICS OF CLASSES

Class declarations

One way to define a class is using a class declaration. To declare a class, you use the class keyword with the name of the class.



```
//  Prototype method example

class Rectangle {
  constructor(height, width) {
    this.height = height;
    this.width = width;
  }
  // Getter
  get area() {
    return this.calcArea();
  }
  // Method
  calcArea() {
    return this.height * this.width;
  }
}

const square = new Rectangle(10, 10);

console.log(square.area); // 100
```

Run the example index.js with node in VSCode

PROJECTS

POP OUT SEARCH

SPLIT LANDING PAGE

FORM VALIDATION

Q&A